# GT 5



## Non-adaptive Algorithms

**Definition** A group testing algorithm is non-adaptive if all tests must be specified without knowing the outcomes of other tests.

Based on this nature, a non-adaptive algorithm can be represented by an $m \times n$ $(0,1)$-matrix if $m$ tests are specified for a group testing on $n$ items. Such a matrix is also known as a pooling design. The example we presented earlier is a $9 \times 12$ $(0,1)$-matrix which can be considered as a pooling design using 9 tests for 12 items. In general, let $M$ be as follows:

$$
M: \quad \begin{array}{c} \\ r_1 \\ r_2 \\ \vdots \\ \vdots \\ r_m \end{array}
\begin{array}{c} c_1\, c_2 \; \cdots \qquad c_n \\ \left[ \begin{array}{l} M(i,j)=1 \text{ if the } j\text{th} \\ \text{item is involved in} \\ \text{the } i\text{th test; and} \\ M(i,j)=0 \text{ otherwise} \end{array} \right] \end{array}
$$

(*) The $j$th item will be represented as a set $C_j$ where $C_j = \{i \mid$ the $j$th item is involved in the $i$th test$\}$.

# GT 5



e.g.

$$M: \begin{array}{c|c} & \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

$C_1 = \{3,4\}$, $C_2 = \{1,3\}$, $C_3 = \{1,2,4\}$, $C_4 = \{3\}$, $C_5 = \{1,2,3,4\}$

(o) Since the items are represented by sets, combining items together can be "seen" as the union of sets.

Definition (d-separable)

M is called d-separable if the unions of d columns are all distinct. If the unions of at most d columns are all distinct, then the matrix is called $\bar{d}$-separable.

Definition (d-disjunct)

M is called d-disjunct if any column does not contain in a union of d (or less) other columns. Equivalently, given any d+1 distinct columns of M, $C_1, C_2, \cdots, C_d, C_{d+1}$, $C_{d+1} \not\subseteq \bigcup_{i=1}^{d} C_i$.

Fact 1. d-disjunct $\Rightarrow$ $\bar{d}$-separable $\Rightarrow$ d-separable

Fact 2. $\overline{d+1}$-separable $\Rightarrow$ d-disjunct

Proof. Consider $C_1, C_2, \cdots, C_{d+1}$. By definition $\bigcup_{i=1}^{d} C_i \neq \bigcup_{j=1}^{d+1} C_j$.

This implies that $C_{d+1} \nsubseteq \bigcup_{i=1}^{d} C_i$. ∎

Following a non-adaptive algorithm, the $m$ tests will be done and an outcome vector $(y_1, y_2, \cdots, y_m)^t$ is obtained. We are aimming at determining the $d$ positive items by using the knowledge on the collection of outcome vectors.

(•) If there are $d$ positive items in a set of $n$ items, then there are $\binom{n}{d}$ different possible combinations.

(•) In order to tell the correct answer, we need at least $\binom{n}{d}$ distinct outcome vectors. Therefore, the minimal requirement of decoding "succefully" is : $2^m \geq \binom{n}{d}$, i.e., $m \geq \lceil \log_2 \binom{n}{d} \rceil$.

Again, this is the informatic lower bound.

(•) If $2^m \geq \binom{n}{d}$, then it's possible to find a $d$-separable matrix which is an $m \times n$ $(0,1)$-matrix. So, far, for general $n$, finding such a matrix with $m \approx \log_2 \binom{n}{d}$ is still <u>far from solved</u>.

Solving a $(d,n)$-problem by using non-adaptive algorithm ($M$: pooling design) can be modeled as follows:

$$M : \binom{N}{d} \xrightarrow{1-1} \{0,1\}^m \qquad (|N| = n)$$

(•) To determine $d$ positives, we need to find the 1-1 correspondence between this vector $(x_1, x_2, \cdots, x_n)^t$ and the outcome vector. Since there are $\binom{n}{d}$ distinct vectors to check, the decoding algorithm takes $O(n^d)$ times to find the answer in the worst case. (This situation comes from the assumption that $M$ is $d$-separable.)

(*) If $M$ is $d$-disjunct, then the decoding is much simpler, it takes $O(m)$ to find all the positive items.

Proof. Let the tests with "0" outcome be the "0-pool", denoted by $t_i(M)$. Assume that $C_1, C_2, \cdots, C_d$ are the positive items. Then, for each negative item $C_{d+1}$, $C_{d+1} \nsubseteq \bigcup_{i=1}^{d} C_i$. Hence, there exists an $x \in C_{d+1}$ such that $x \notin \bigcup_{i=1}^{d} C_i$. If $x$ is in the $i$th test, $t_i$, then $t_i \in t^-(M)$. Hence, $C_{d+1}$ can be found by using this test $t_i$.

In fact, all items involved in the ith test are all negative!
Since checking all the 0-pools in $t^{-}(M)$ can find all negative items, all positive items can be found subsequently. ∎

A quick review about the $9\times12$ pooling design mentioned in previous lecture is in fact a 2-disjunct matrix and therefore we can use it to find $d \le 2$ positive items with fairly quick decoding algorithm. Similar idea from combinatorial design (Projective plane of order 3) can be applied to find other disjunct matrices. Here we present the first systematic construction.

Theorem (A.J. Macula, DM 1996)

Let $M$ be indexed by $\binom{n}{d}$ rows from $\binom{N}{d}$ and $\binom{n}{k}$ columns from $\binom{N}{k}$ where $N=[1,n]$ and $d\le k$. Furthermore, let $M(i,j) = 1$ if the $i^{th}$ $d$-subset of $N$ is contained in the $j^{th}$ $k$-subset of $N$ and 0 otherwise. Then, $M$ is a $d$-disjunct matrix.

Proof. Let $C_1, C_2, \cdots, C_d, C_{d+1}$ be any $d+1$ columns. It suffices to find one row, say the $i^{th}$ row, such that $M(i,d+1)=1$ but $M(i,j)=0$ for $j=1,2,\cdots,d$.

Since $C_1, C_2, \cdots, C_{d+1}$ are distinct $k$ subsets of $N$, $C_{d+1} \setminus C_j \neq \phi$ for $j = 1, 2, \cdots, d$. Let $x_j \in C_{d+1} \setminus C_j$ and $D = \{x_1, x_2, \cdots, x_j\}$ (a multi-set). Now, clearly $D \subseteq C_{d+1}$ but $D \nsubseteq C_j$ for each $j = 1, 2, \cdots, d$. Let $\tilde{D}$ be a $d$-subset of $N$ such that $D \subseteq \tilde{D} \subseteq C_{d+1}$ and let $\tilde{D}$ occur in the $i^{th}$ row. Since $C_1, C_2, \cdots, C_d$ can not contain $\tilde{D}$, the proof follows. ∎

(•) We remark here that a good pooling design (for $(d, n)$-problem) should have two good properties:

(1) $\dfrac{\text{the \# of items}}{\text{the \# of tests}}$ is large; and

(2) fast decoding algorithm.

(••) The above properties are not the only requirements for group testings with extra "challenges", say the occurrence of errors, inhibitors, ..., etc.